Embedded Intel® Architecture Design Solutions for Network Caching

Krishna Shetty
Applications Architect
Embedded Intel Architecture Division
Intel Corporation

Dave Hillyard
Platforms Manager
Embedded Intel Architecture Division
Intel Corporation

Embedded Intel® Architecture Design Solutions for Network Caching

Contents

Executive Summary

The increasing popularity of the Internet is driving explosive growth in data networking. While the Web supports a vast amount of data traffic, a large percentage of this traffic is redundant. This is due to the fact that multiple users at any given location typically request much of the same information at any given time. As a result, a significant portion of Wide Area Network (WAN) traffic consists of identical request and data content. Web caching is a technique designed to optimize available bandwidth and improve overall network performance by storing frequently accessed information closer to the user. In addition to its network performance benefits, Web caching provides an opportunity to reduce network communication costs.

The Internet community became aware of the benefits of Web caching long before the Web evolved to become a commercial phenomenon. Internet protocols such as FTP, Gopher, and news groups are mirrored around the world in order to bring popular files closer to users. With the advent of HTTP, mirroring of content became less practical, due to the high volume, time-sensitive and random nature of the content being requested.

Intel supports caching at all levels of the network infrastructure with an integrated reference design including CPUs, chipsets, flash memory and appropriate network interface solutions. Several vendors are developing caching appliances based on Intel® Architecture hardware, readily available software, and standardized interface components to achieve substantial cost savings and shorter time-to-market.

The Intel Cache Appliance Reference Design

Figure 1 is a block diagram of a typical appliance board based on embedded Intel® Architecture components including an Intel® Celeron™ processor, Intel® 440BX chipset (including the system controller 82443BX and PIIX4E South Bridge), the Intel® 82559 Ethernet controller with Intel® Flash memory and SDRAM components. This board provides a fast time-to-market development platform that can be used as a stand-alone integrated Internet appliance for caching applications.

The Celeron processor shown has an operating frequency of 300 MHz and higher with a built-in 128 Kbytes L2 cache running at the processor's full core speed. The processor's internal cache satisfies most of the bandwidth and latency requirements of the CPU core. The L2 cache reduces the CPU's external memory bandwidth requirement and makes the processor's performance less sensitive to bus access latency. Although Intel has developed the built-in cache and the proprietary bus between the CPU and the cache to enhance PC-specific applications, it has been found to be enormously advantageous in raising the performance of communications functions including routing table look-up. The Celeron processor features a dynamic execution microarchitecture and also executes Intel® MMX™ technology instructions for enhanced media and communications performance.

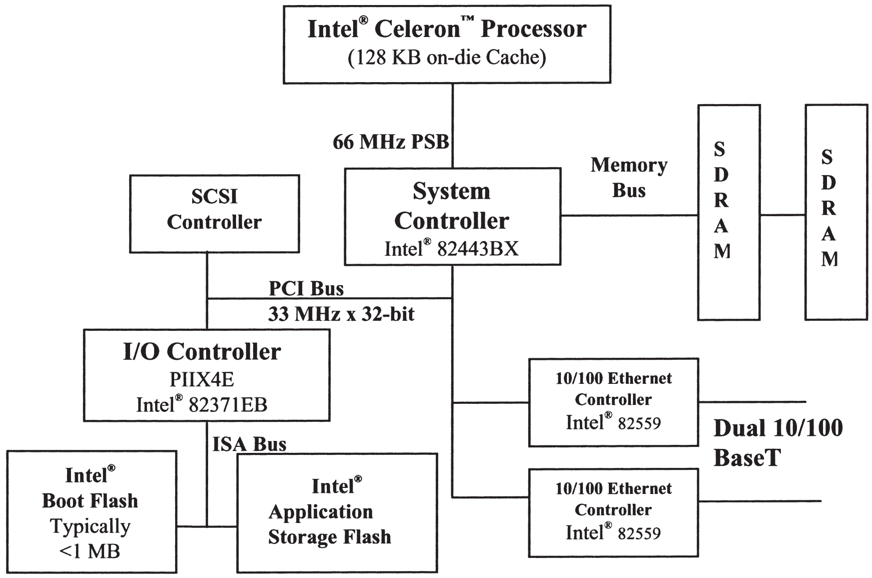

Figure 1 - Typical Network Cache Appliance - Functional Block Diagram

The system controller can address up to 1 Gbyte of memory, which is adequate in many communications applications. In the network caching context, this is enough memory to support several million objects in the hard drive, while supporting more than 2,000 concurrent clients. Although SCSI drives are generally used in caching applications, the system shown in the figure can be connected to IDE drives too. The IDE controller built into the PIIX4E can support up to four drives with an Ultra DMA/33 interface. When SCSI drives are used, PCI-compatible SCSI controllers are needed, as shown in the figure.

The high performance chipset from Intel contains 82443BX Host to PCI Bridge and PIIX4E South Bridge. The Processor System Bus (PSB) between the CPU and chipset operates at 66 MHz and supports both in-order and dynamic deferred transactions. The pipelined PSB bus architecture is tuned for multiprocessing and is capable of eight outstanding transactions. It has been observed in many communications applications that a balanced match between the processor and I/O performance can be more critical to performance than the speed of the CPU alone. The pipelined structure on the PSB is designed to balance I/O performance with CPU speed. It is proving to be extremely useful in supporting concurrent transactions found in many communication applications.

Caching

Caching is not a new concept. For years computers have used caching to improve system performance by storing frequently used information closer to the processor. At the network level, caching can be defined as a technique for keeping frequently accessed information in a location closer to the requester. Because of its physical or logical proximity to the user, the cache storage location can provide faster access, compared to retrieving data from across the network.

All cache servers work basically the same way. They intercept object requests traveling from a browser to a Web server, and then store the requests on a hard drive as the objects travel back from the Web server to the browser. In this way, subsequent requests for the same object by other browsers can be intercepted by the cache, which then returns the object from its own memory rather than forwarding the request to a remote server. Ideally, object requests fulfilled by the cache should save both time and communications bandwidth. Caching can reduce or eliminate a number of key factors that can cause delays, including:

- Transmission of a request from the end-user to the original storage device.
- Network congestion between the end-user and the original storage device.
- Congestion within the storage device itself.
- Transmission of the data from the storage device back to the end-user.

Client Caching

Browser applications allow an individual to cache Web pages (images and HTML text) on the client's local hard drive. The ability to cache Web objects at the browser has been available for some time. The user can configure the amount of disk space devoted to caching. This method serves single users well, but it provides no benefit for other users on the same network who may be accessing the same Web sites.

Network Caching

In order to optimize the use of available network bandwidth and benefit multiple users, the caching concept has been extended to the network level. For example, network caching is currently widely used by Internet Service Providers (ISPs) who wish to increase the level of their service. On the consumer side, faster connectivity solutions such as Digital Subscriber Lines (DSL) and cable modems are enhancing Web browsing performance, which is in turn increasing the traffic load on the network backbone. At the same time, content providers are diversifying and enriching data types with more complex formats such as streaming audio/video and Java applets.

These richer data types tend to increase the bandwidth requirement of each individual Internet session. These trends are forcing ISPs to look for cost-effective ways to enhance the existing communications and networking infrastructure, while keeping pace with customer demand. Network caching will continue to be a big part of their solution.

Network Caching Market

Although ISPs realize the need for caching, enterprises have yet to adopt the technology on a large scale. However, market analysts expect caching to saturate the enterprise network. According to Collaborative Research, the growth of caching in enterprises should rapidly overtake that of ISPs, growing from $85 million in 1998 to over $1 billion in 2000 and $3 billion in 2002.

Cache solution vendors who have supplied high-end products to ISPs now realize the growing need for caching in enterprise networks and are responding by positioning their products at the lower end of the spectrum to meet the needs of SOHO (Small Office Home Office), workgroup and enterprise environments. Several cache solution vendors are already developing appliances based on industry-standard hardware, readily available software, and standardized interface components to achieve substantial cost savings and shorter time-to-market.

Advantages and Disadvantages of Network Caching

The primary rationale for incorporating caching technology into enterprise and ISP networks is to make effective use of available bandwidth and to improve the performance of browser-based applications. Network caching is the most cost-effective way to address the bandwidth bottlenecks that inevitably result from the increased use of the Internet within an organization. Merely increasing the bandwidth of WAN connections to remote sites may not always lead to the desired solution. The reason for this is that the latency between a request and the delivery of a Web object involves a combination of several factors, including the speed at which the signals travel over the medium, congestion at the server, and physical delays associated with the network nodes, transmission system, and routers.

The efficiency of a cache scales with the hit rate. The cache needs to acquire the first copy of the object and then store it, and the number of objects stored will determine the hit rate. The number of users, as well as the statistical sample of users who will try to access the same Web page at any given time also dictates the hit rate.

Worldwide Internet Caching Market: 2000-2002 (in Million Dollars)

Segment	2000	2001	2002
Enterprise	1,113	2,108	3,157
ISPs	376	481	576
Other	149	259	373
Total	1,638	2,848	4,106

Source: Collaborative Research

Larger community sizes tend to produce better quality statistics and higher cache hit rates.

Caching also has its drawbacks. The current trend on the part of online content providers is to dynamically deliver constantly refreshed content on the Web. Content providers can be legitimately concerned that visitors may receive stale information from a cached Web site. The response to this concern is that modern caches typically include provisions that prevent the caching of dynamic pages. Even if all dynamically generated HTML documents were to become uncacheable, caching would still improve Web performance. Research indicates that static objects, such as navigation icons, buttons, and company logos may represent 70 to 80 percent of Web traffic. These static objects can be cached to improve overall Web access performance.

Another disadvantage to caching is that content providers would not be able to get actual hit-count statistics, preventing them from reliably measuring Web traffic to their sites. In cases where commercial sites rely on accurate visitor logs to configure and provision their content, this concern is genuine. In addition, calculating advertising revenue will be impaired by inaccurate hit counts. Several innovative techniques have been put forward to overcome this shortcoming, including the incorporation of a hit-reporting feature to forward hit counts to a Web server for statistical purposes, after an object is delivered from its local cache.

Measuring Performance

The critical determinants of caching performance are disk I/O, Transmission Control Protocol (TCP) stack characteristics, and caching routines. To improve caching performance and reliability, each of these areas must be optimized. There are typically three ways to measure caching performance: hit ratio, byte ratio, and average response time. Hit ratio compares the number of hits fulfilled by the cache to the total number of objects requested. Similarly, byte ratio compares the size of the objects delivered from the cache to the size of all objects delivered. The third measurement, average response time, relates to the comparison of response times of objects retrieved directly from the Internet without caching to the response times for the same objects delivered by the cache. While hit ratio and byte ratio are easily calculated from the log files maintained by cache servers, response time is subjective and can only be approximated in a real-world environment.

Other factors that can impact caching performance include the size of the user community and the size of the cache itself. Research by the National Laboratory for Applied Network Research (NLANR), a consortium of National Science Foundation (NSF)-supported supercomputer sites established in 1995, indicates that a hit ratio of 30 percent is relatively easy to achieve, while rates of 50 percent or higher are possible with larger communities or with a larger cache. NLANR also found that byte ratios tend to be about 10 percent lower than hit ratios, which indicates that smaller objects are more commonly cached than larger ones.

Network Caching Methods

Network caching can be implemented in two ways: as a proxy or in transparent mode.

Proxy Caching
Early caches were typically implemented on general-purpose servers running proxy caching as an application, along with many other applications. Proxy caches are intrusive in accepting requests and passing them on to their destinations. Every Web client must be configured to direct all outgoing calls to the proxy's IP address. The proxy checks if it has cached the valid reply for a requested object. If it has not cached the valid reply, the proxy makes a decision to forward the request to the destination server. In addition, a proxy may also implement security firewall, user authentication, traffic administration, content filtering, and a host of other functions. The proxy cache also allows easy implementation of security and control.

Although proxy caches help in value-added services, they slow down Web access, and any failure of the proxy server causes all users to lose network access. The cache server can also be overloaded and add an incremental performance limitation. Further, proxies require configuration of each user's browser, and they lack scalability.

The earliest proxy cache is the Harvest cache originated by the Advanced Research Project Agency (ARPA), the

NSF and National Aeronautics and Space Administration (NASA). These research efforts have quickly evolved into several commercial cache offerings for the enterprise market segment, along with other software packages like security, user authentication, virus scanning etc.

Over time, browser set-up and support will be completely automated. A typical browser will automatically find whatever resources it needs, including the cache, each time it begins operation. At that point, proxy caching will be completely transparent to the user. Until that day comes, transparency issues are a key inhibitor to the use of proxy caches.

Transparent Caching

A transparent cache hangs off the network near the Internet gateway and invisibly intercepts the traffic as it passes from the browser to the Web server. The transparent cache uses a policy-based router or a layer-4 switch that diverts the HTTP traffic to the cache server or a group of servers. A truly transparent caching solution should be flexible, support scalability, and incorporate adequate load balancing among multiple cache servers. It should also be working in 'fail safe' mode in the event of one or all the cache servers become unavailable. Transparent caching eliminates the need to configure individual browsers to point to the cache, while reducing administrative overhead.

To provide benefits to users, the transparent cache must be placed at a "hand-off point" in the network through which all the traffic passes. Using a transparent cache therefore requires a complete understanding of the traffic routing in the network in order to identify the right location to situate the cache.

Caching Protocols

A number of caching protocols are prominent in the industry. They include: Hyper Text Transfer protocol (HTTP/1.0 and HTTP/1.1), Internet Cache Protocol (ICP), Cache Array Routing Protocol (CARP), and Web Cache Control Protocol (WCCP).

- HTTP is used to transfer data between Web servers (e.g. Apache*) and Web clients (e.g. Netscape* Navigator). Currently the majority of clients and servers use HTTP/1.0, although deployment of HTTP/1.1 has already begun.

- Internet Caching Protocol (ICP), which sprang from the early caching research of the Harvest project, is a prominent cache cluster protocol that provides a quick and efficient method of inter-cache communication. It offers a mechanism for establishing complex cache hierarchies. Through ICP, one can configure a cache to query other caches that also support ICP to determine if they contain recent information about Web objects.

  For example, a local cache can poll an upstream cache to see if it has retrieved a more recent copy of a file, or if not, whether it has verified the age of the file at the source server. Even if the upstream server hasn't retrieved a newer version of the file, it may have more recently verified information that the file at the source hasn't been updated. Depending on the refreshing algorithm of the local cache, the local cache may use this information to obtain a newer version of the object from the source server, or use the local copy instead. Polling an upstream cache creates more latency because the round-trip distance and time are increased. Regardless, there may still be a significant timesaving since the request does not need to travel all the way to the object's source server. In addition, serving objects from ICP-linked caches can reduce congestion on the network backbone, freeing up bandwidth for the Internet user population in general. Nearly all caches on the market today support ICP. ICP messages are usually small (median 66 bytes) and transmitted as UDP packets. They consist of a 20-byte header followed by a URL string. In a cache hierarchy, there will usually be many more ICP messages than HTTP requests. Depending on the type of relationship (parent or sibling) the ratio of ICP to HTTP messages may be anywhere from 2:1 to 10:1.

- Caching Array Routing Protocol (CARP) is a peer-to-peer protocol that is primarily concerned with the sharing of cache load within a local server farm. It was developed by Microsoft*, and has been submitted to the World Wide Web Consortium (W3C) as an Internet draft proposal.

- To enable its Cache Engine appliance to operate with its routers, Cisco developed the Web Cache Communication Protocol (WCCP). Using WCCP, a Cisco* IOS router intercepts HTTP requests coming from browsers and redirects them to a cache server or appliance. WCCP supports scalability by distributing requests to multiple cache servers based on availability. Although a proprietary development, Cisco began licensing the WCCP to other cache vendors in November 1998.

Cache Hierarchy

Cache hierarchies are a logical extension of the caching concept. A group of network caches can benefit from sharing another cache in the same way that a group of network clients can benefit from sharing a cache. A cache hierarchy is a collection of caches that is organized in a logical parent/child and sibling arrangement so that caches closest to the Internet backbone or gateways act as parents to caches located farthest from the backbone. The parent cache resolves the 'misses' for their children. This ensures that the cache hierarchy achieves the maximum reduction in bandwidth utilization on the backbone links. It also helps to reduce the load on the source servers. This technique also helps to build rich cache content on the parent caches so that other child caches in the hierarchy obtain better hit-rates. In addition to parent and child caches, there can be sibling caches. These are caches at the same level in the hierarchy.

Each cache in the hierarchy independently uses a simple resolution protocol to decide whether to fetch the reference from the object's home site or from parent or sibling caches. Siblings will not generally fetch an object for another sibling to resolve a cache 'miss'.

Whether the advantages of hierarchical caching outweigh the disadvantages depends on the specific situation.

The main advantages are:

- Additional cache hits. In general, an additional 10 percent of requests received by the cache would be cache hits in the neighbor caches.
- By routing requests to selective caches, it may be possible to off-load busy HTTP traffic along a certain cost-effective path.

Among the disadvantages:

- Configuring neighbor caches requires coordination, a burden that gets worse as membership grows.
- There is an additional delay for cache misses. Whether utilizing a neighbor cache improves end-user performance depends on many factors, including delays between peers and link congestion.

Intel and Cache Appliances

Intel supports caching at all levels of network infrastructure with an integrated reference design that includes integrated Intel Architecture processors, chipsets, flash memory and appropriate network interface solutions. By providing these components in a compact form factor, Intel is making the PC technology available to developers of cache appliances at affordable price points. The extended life cycle of these Intel components is another factor that will accelerate the convergence of PC and communications technologies. The technology convergence is further supported by the ability of caching to enhance network performance by optimizing the use of communication bandwidth to match the increasing speed and performance of connected PCs.

Benefits for Developers

Intel's communications reference design has significant advantages for developers:

- It can dramatically accelerate time-to-market with a comprehensive hardware reference platform. Schematics are available for download at no cost from Intel's Developer Site: developer.intel.com/platforms/applied/comm/index.htm#configs.
- The reference design is based on open Intel Architecture that is familiar to most programmers, and the architecture supports multiple operating systems. Scalable embedded Intel Architecture enables appliance developers to differentiate products through software-based value-added features and functionality, while maintaining the levels of performance that end-users expect.
- Embedded Intel Architecture encompasses a large variety of development tools and software, including the components, tools and software available for the PC developers. Board-level solutions are supported by a variety of third-party vendors.
- The Intel roadmap leads to higher levels of performance. Intel processors and other components are designed to meet the embedded lifecycle requirements of communications applications.
- Intel manufacturing capacity and quality helps ensure product reliability and customer satisfaction.

Intel® Internet Exchange Architecture

The Intel® Internet Exchange architecture (IXA), as shown in Figure 2, provides a consistent framework for OEMs and independent software vendors to quickly deploy new networking and communications services and develop differentiated networking products that deliver scalable performance with reduced total cost of ownership. Intel IXA includes end-to-end development solutions and building blocks that enable developers to create solutions for the entire Open System Interconnectivity (OSI) stack.

Embedded Intel Architecture delivers solutions that meet the performance requirements of the Application Services Layer of the OSI Model. By incorporating scalable embedded Intel Architecture components and software within Intel IXA, Intel is delivering a flexible top-to-bottom architecture that delivers high performance, scalability, code compatibility and programmability that enables faster and more cost effective software-based product differentiation.

Conclusion

With the convergence of voice and data and increased use of the Internet in enterprise network environments, work groups, and small office/home office settings, network caching is becoming a powerful and cost-effective tool to improve network performance, without unduly disturbing the existing communications infrastructure. Adding cache appliances to the existing network at critical points can reduce network latency and improve access efficiency, while keeping WAN implementation costs under control. The stand-alone nature of the cache appliance, easy installation, maintenance, and configuration are other features that make them attractive in enhancing the value of the Internet.

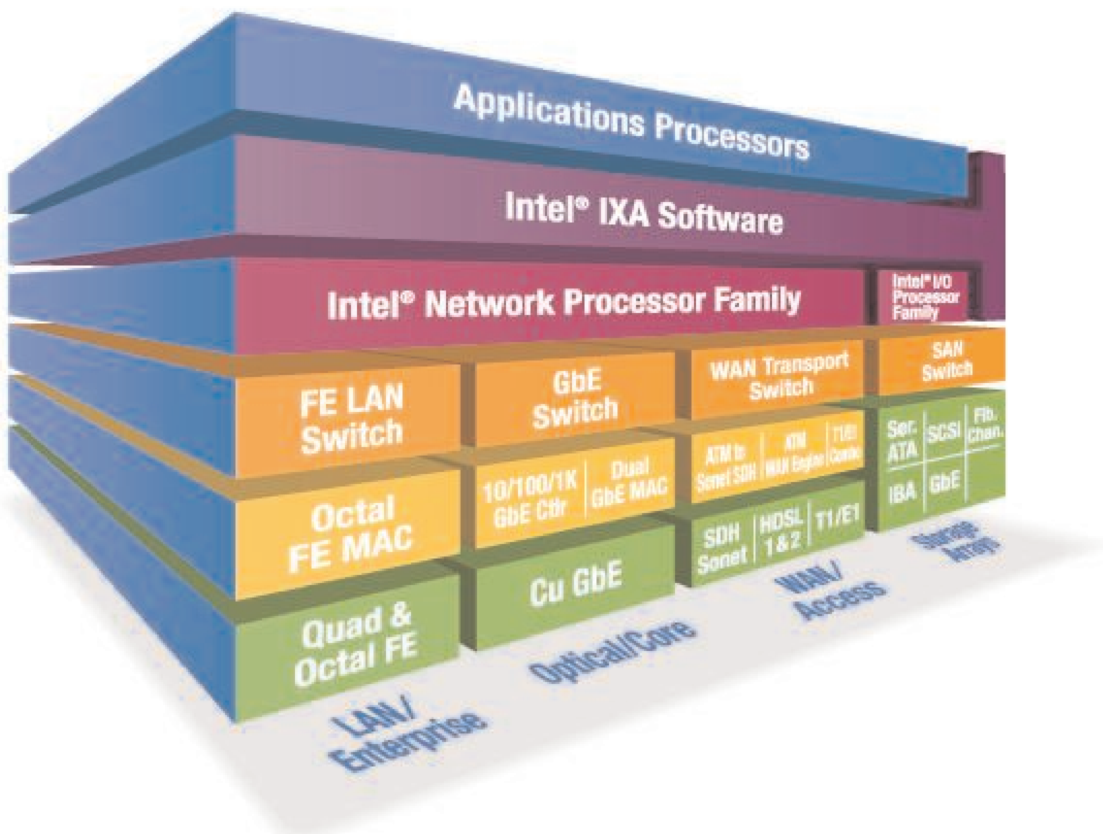

Figure 2 - Intel® Internet Exchange Architecture

For More Information

For more information on Intel's reference designs for communications, visit:	developer.intel.com/platforms/applied/comm
For additional information on the Intel Celeron processor, visit:	developer.intel.com/design/celeron/datashts/243658.htm
For additional information on the Intel 440BX chipset, visit:	developer.intel.com/design/chipsets/440bx/index.htm
Details on the Intel 82559 Ethernet controller are available at:	developer.intel.com/design/network/82559.htm
For more information on Intel Internet Exchange Architecture solutions, visit:	www.intel.com/ixa

Bibliography

CacheFlow White papers, http://www.cacheflow.com/technology/wp/activecaching.html

Danzig, Peter, *"NetCache Architecture and Deployment"*, http://www.netapp.com/technology/level3/3029.html

Introduction to Web Caching, http://cosmos.kaist.ac.kr/cs540/course-archive/981019/sld001.htm

Keppeler, Phil., *"A Little Cache Goes a Long Way"*, Network Magazine, December 1998

Keppeler, Phil, *"Cashing In on IP Caching"*, Network Magazine, February 1999

Packetstorm Technologies; http://packetstorm.on.ca/products/webspeed

Web Proxy Caching, http://www.cs.duke.edu/education/courses/cps212/fall98/slides-html/webcache/sld001.htm

Order Number: 273380-001 Printed in USA/1000/7.5K/IL2110F LY